PRACTICAL 1
COMMANDS

1. tar command examples

Create a new tar archive.

```
$ tar cvf archive_name.tar dirname/
```

Extract from an existing tar archive.

```
$ tar xvf archive_name.tar
```

View an existing tar archive.

```
$ tar tvf archive_name.tar
```

2. grep command examples

Search for a given string in a file (case in-sensitive search).

```
$ grep -i "the" demo_file
```

Print the matched line, along with the 3 lines after it.

```
$ grep -A 3 -i "example" demo_text
```

Search for a given string in all files recursively

```
$ grep -r "ramesh" *
```

3. find command examples

Find files using file-name (case in-sensitve find)

```
# find -iname "MyCProgram.c"
```

Execute commands on files found by the find command

```
$ find -iname "MyCProgram.c" -exec md5sum {} \;
```

Find all empty files in home directory

```
# find ~ -empty
```

4. sed command examples

Print file content in reverse order

```
$ sed -n '1!G;h;$p' thegeekstuff.txt
```

Add line number for all non-empty-lines in a file

```
$ sed '/./=' thegeekstuff.txt | sed 'N; s/\n/ /'
```

5. awk command examples

Remove duplicate lines using awk

```
$ awk '!($0 in array) { array[$0]; print }' temp
```

Print all lines from /etc/passwd that has the same uid and gid

```
$awk -F ':' '$3==$4' passwd.txt
```

Print only specific field from a file.

```
$ awk '{print $2,$5;}' employee.txt
```

6. sort command examples

Sort a file in ascending order

```
$ sort names.txt
```

Sort a file in descending order

```
$ sort -r names.txt
```

Sort passwd file by 3rd field.

```
$ sort -t: -k 3n /etc/passwd | more
```

7. ls command examples

Display filesize in human readable format (e.g. KB, MB etc.,)

```
$ ls -lh
-rw-r----- 1 ramesh team-dev 8.9M Jun 12 15:27 arch-linux.txt.gz
```

Order Files Based on Last Modified Time (In Reverse Order) Using ls -ltr

```
$ ls -ltr
```

Visual Classification of Files With Special Characters Using ls -F

```
$ ls -F
```

8. pwd command

pwd is Print working directory. What else can be said about the good old pwd who has been printing the current directory name for ages.

9. cd command examples

Use “cd -” to toggle between the last two directories

10. gzip command examples

To create a *.gz compressed file:

```
$ gzip test.txt
```

To uncompress a *.gz file:

```
$ gzip -d test.txt.gz
```

Display compression ratio of the compressed file using gzip -l

```
$ gzip -l *.gz
         compressed        uncompressed  ratio uncompressed_name
              23709               97975  75.8% asp-patch-rpms.txt
```

11. bzip2 command examples

To create a *.bz2 compressed file:

```
$ bzip2 test.txt
```

To uncompress a *.bz2 file:

```
bzip2 -d test.txt.bz2
```

12. unzip command examples

To extract a *.zip compressed file:

```
$ unzip test.zip
```

View the contents of *.zip file (Without unzipping it):

```
$ unzip -l jasper.zip
Archive:  jasper.zip
  Length     Date   Time    Name
 --------    ----   ----    ----
    40995  11-30-98 23:50   META-INF/MANIFEST.MF
    32169  08-25-98 21:07   classes_
    15964  08-25-98 21:07   classes_names
    10542  08-25-98 21:07   classes_ncomp
```

13. shutdown command examples

Shutdown the system and turn the power off immediately.

```
# shutdown -h now
```

Shutdown the system after 10 minutes.

```
# shutdown -h +10
```

Reboot the system using shutdown command.

```
# shutdown -r now
```

Force the filesystem check during reboot.

```
# shutdown -Fr now
```

14. service command examples

Service command is used to run the system V init scripts. i.e Instead of calling the scripts located in the /etc/init.d/ directory with their full path, you can use the service command.

Check the status of a service:

```
# service ssh status
```

Check the status of all the services.

```
service --status-all
```

Restart a service.

```
# service ssh restart
```

15. ps command examples

ps command is used to display information about the processes that are running in the system.

While there are lot of arguments that could be passed to a ps command, following are some of the common ones.

To view current running processes.

```
$ ps -ef | more
```

To view current running processes in a tree structure. H option stands for process hierarchy.

```
$ ps -efH | more
```

16. free command examples

This command is used to display the free, used, swap memory available in the system.

Typical free command output. The output is displayed in bytes.

```
$ free
             total       used       free     shared    buffers     cached
Mem:       3566408    1580220    1986188          0     203988     902960
-/+ buffers/cache:     473272    3093136
Swap:      4000176          0    4000176
```

If you want to quickly check how many GB of RAM your system has use the -g option. -b option displays in bytes, -k in kilo bytes, -m in mega bytes.

```
$ free -g
             total       used       free     shared    buffers     cached
Mem:             3          1          1          0          0          0
-/+ buffers/cache:          0          2
Swap:            3          0          3
```

If you want to see a total memory (including the swap), use the -t switch, which will display a total line as shown below.

```
ramesh@ramesh-laptop:~$ free -t
             total       used       free     shared    buffers     cached
Mem:       3566408    1592148    1974260          0     204260     912556
-/+ buffers/cache:     475332    3091076
Swap:      4000176          0    4000176
Total:     7566584    1592148    5974436
```

17. top command examples

top command displays the top processes in the system (by default sorted by cpu usage). To sort top output by any column, Press O (upper-case O) , which will display all the possible columns that you can sort by as shown below.

```
Current Sort Field:  P  for window 1:Def
Select sort field via field letter, type any other key to return

  a: PID        = Process Id             v: nDRT       = Dirty Pages count
  d: UID        = User Id                y: WCHAN      = Sleeping in Function
  e: USER       = User Name              z: Flags      = Task Flags
  ........
```

To displays only the processes that belong to a particular user use -u option. The following will show only the top processes that belongs to oracle user.

```
$ top -u oracle
```

18. df command examples

Displays the file system disk space usage. By default df -k displays output in bytes.

```
$ df -k
Filesystem           1K-blocks      Used Available Use% Mounted on
/dev/sda1             29530400   3233104  24797232  12% /
/dev/sda2            120367992  50171596  64082060  44% /home
```

df -h displays output in human readable form. i.e size will be displayed in GB’s.

```
ramesh@ramesh-laptop:~$ df -h
Filesystem            Size  Used Avail Use% Mounted on
/dev/sda1              29G  3.1G   24G  12% /
/dev/sda2             115G   48G   62G  44% /home
```

Use -T option to display what type of file system.

```
ramesh@ramesh-laptop:~$ df -T
Filesystem    Type   1K-blocks      Used Available Use% Mounted on
/dev/sda1     ext4    29530400   3233120  24797216  12% /
/dev/sda2     ext4   120367992  50171596  64082060  44% /home
```

19. kill command examples

Use kill command to terminate a process. First get the process id using ps -ef command, then use kill -9 to kill the running Linux process as shown below. You can also use killall, pkill, xkill to terminate a unix process.

```
$ ps -ef | grep vim
ramesh    7243  7222  9 22:43 pts/2    00:00:00 vim

$ kill -9 7243
```

20. rm command examples

Get confirmation before removing the file.

```
$ rm -i filename.txt
```

It is very useful while giving shell metacharacters in the file name argument.

Print the filename and get confirmation before removing the file.

```
$ rm -i file*
```

Following example recursively removes all files and directories under the example directory. This also removes the example directory itself.

```
$ rm -r example
```

21. cp command examples

Copy file1 to file2 preserving the mode, ownership and timestamp.

```
$ cp -p file1 file2
```

Copy file1 to file2. if file2 exists prompt for confirmation before overwritting it.

```
$ cp -i file1 file2
```

22. mv command examples

Rename file1 to file2. if file2 exists prompt for confirmation before overwritting it.

```
$ mv -i file1 file2
```

Note: mv -f is just the opposite, which will overwrite file2 without prompting.

mv -v will print what is happening during file rename, which is useful while specifying shell metacharacters in the file name argument.

```
$ mv -v file1 file2
```

23. cat command examples

You can view multiple files at the same time. Following example prints the content of file1 followed by file2 to stdout.

```
$ cat file1 file2
```

While displaying the file, following cat -n command will prepend the line number to each line of the output.

```
$ cat -n /etc/logrotate.conf
    1   /var/log/btmp {
```

```
    2       missingok
    3       monthly
    4       create 0660 root utmp
    5       rotate 1
    6   }
```

24. chmod command examples

chmod command is used to change the permissions for a file or directory.

Give full access to user and group (i.e read, write and execute) on a specific file.

```
$ chmod ug+rwx file.txt
```

Revoke all access for the group (i.e read, write and execute) on a specific file.

```
$ chmod g-rwx file.txt
```

Apply the file permissions recursively to all the files in the sub-directories.

```
$ chmod -R ug+rwx file.txt
```

25. passwd command examples

Change your password from command line using passwd. This will prompt for the old password followed by the new password.

```
$ passwd
```

Super user can use passwd command to reset others password. This will not prompt for current password of the user.

```
# passwd USERNAME
```

Remove password for a specific user. Root user can disable password for a specific user. Once the password is disabled, the user can login without entering the password.

```
# passwd -d USERNAME
```

26. mkdir command examples

Following example creates a directory called temp under your home directory.

```
$ mkdir ~/temp
```

Create nested directories using one mkdir command. If any of these directories exist already, it will not display any error. If any of these directories doesn’t exist, it will create them.

```
$ mkdir -p dir1/dir2/dir3/dir4/
```

27. ifconfig command examples

Use ifconfig command to view or configure a network interface on the Linux system.

View all the interfaces along with status.

```
$ ifconfig -a
```

Start or stop a specific interface using up and down command as shown below.

```
$ ifconfig eth0 up

$ ifconfig eth0 down
```

28. uname command examples

Uname command displays important information about the system such as — Kernel name, Host name, Kernel release number,
Processor type, etc.,

Sample uname output from a Ubuntu laptop is shown below.

```
$ uname -a
Linux john-laptop 2.6.32-24-generic #41-Ubuntu SMP Thu Aug 19 01:12:52 UTC 2010
i686 GNU/Linux
```

29. whereis command examples

When you want to find out where a specific Unix command exists (for example, where does ls command exists?), you can execute the following command.

```
$ whereis ls
ls: /bin/ls /usr/share/man/man1/ls.1.gz /usr/share/man/man1p/ls.1p.gz
```

30. whatis command examples

Whatis command displays a single line description about a command.

```
$ whatis ls
ls              (1)  - list directory contents

$ whatis ifconfig
ifconfig (8)         - configure a network interface
```

31. locate command examples

Using locate command you can quickly search for the location of a specific file (or group of files). Locate command uses the database created by updatedb.

The example below shows all files in the system that contains the word crontab in it.

```
$ locate crontab
/etc/anacrontab
/etc/crontab
/usr/bin/crontab
/usr/share/doc/cron/examples/crontab2english.pl.gz
/usr/share/man/man1/crontab.1.gz
/usr/share/man/man5/anacrontab.5.gz
/usr/share/man/man5/crontab.5.gz
/usr/share/vim/vim72/syntax/crontab.vim
```

32. man command examples

Display the man page of a specific command.

```
$ man crontab
```

When a man page for a command is located under more than one section, you can view the man page for that command from a specific section as shown below.

```
$ man SECTION-NUMBER commandname
```

Following 8 sections are available in the man page.

1. General commands
2. System calls
3. C library functions
4. Special files (usually devices, those found in /dev) and drivers
5. File formats and conventions
6. Games and screensavers
7. Miscellaneous
8. System administration commands and daemons

For example, when you do whatis crontab, you'll notice that crontab has two man pages (section 1 and section 5). To view section 5 of crontab man page, do the following.

```
$ whatis crontab
crontab (1)          - maintain crontab files for individual users (V3)
crontab (5)          - tables for driving cron

$ man 5 crontab
```

33. tail command examples

Print the last 10 lines of a file by default.

```
$ tail filename.txt
```

Print N number of lines from the file named filename.txt

```
$ tail -n N filename.txt
```

View the content of the file in real time using tail -f. This is useful to view the log files, that keeps growing. The command can be terminated using CTRL-C.

```
$ tail -f log-file
```

34. less command examples

less is very efficient while viewing huge log files, as it doesn't need to load the full file while opening.

```
$ less huge-log-file.log
```

One you open a file using less command, following two keys are very helpful.

```
CTRL+F – forward one window
CTRL+B – backward one window
```

35. yum command examples

To install apache using yum.

```
$ yum install httpd
```

To upgrade apache using yum.

```
$ yum update httpd
```

To uninstall/remove apache using yum.

```
$ yum remove httpd
```

36. ping command examples

Ping a remote host by sending only 5 packets.

```
$ ping -c 5 gmail.com
```

37. date command examples

Set the system date:

```
# date -s "01/31/2010 23:59:53"
```

Once you've changed the system date, you should syncronize the hardware clock with the system date as shown below.

```
# hwclock –systohc
```

```
# hwclock --systohc –utc
```

SIMPLE C PROGRAM

Open Ubuntu's graphical Text Editor and write or copy the following sample program into it:

```
#include<stdio.h>

int main()
{
printf("\nA sample C program\n\n");
return 0;
}
```

Then save the file with .c extension. In this example, I am naming my C program as sampleProgram.c

Alternatively, you can write the C program through the Terminal in gedit as follows:

```
$ gedit sampleProgram.c
```

This will create a .c file where you can write and save a program.

Compile the C program with gcc

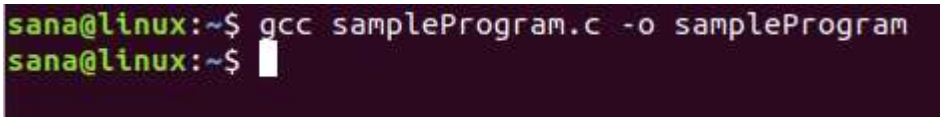
In your Terminal, enter the following command in order to make an executable version of the program you have written:

Syntax:

```
$ gcc [programName].c -o programName
```

Example:

```
$ gcc sampleProgram.c -o sampleProgram
```



Make sure your program is located in your Home folder. Otherwise, you will need to specify appropriate paths in this command.

Run the program

The final step is to run the compiled C program. Use the following syntax to do so:

```
$ ./programName
```

Example:

```
$ ./sampleProgram
```